Postel

RFC # 733
NIC # 41952

Obsoletes: RFC #561 (NIC #18516)
RFC #680 (NIC #32116)
RFC #724 (NIC #37435)

STANDARD FOR THE FORMAT OF ARPA NETWORK TEXT MESSAGES(1)

21 November 1977

by

David H. Crocker
The Rand Coproration

John J. Vittal
Bolt Beranek and Newman Inc.

Kenneth T. Pogran
Massachussets Institute of Technology

D. Austin Henderson, Jr.(2)
Bolt Beranek and Newman Inc.

(1)This work was supported by the Defense Advanced Research Projects Agency of the Department of Defense, under contract Nos. N00014-75-C-0661, MDA903-76-C-0212, and DAHC15-73-C0181.

(2)The authors' postal addresses are: D. Crocker, The Rand Corporation, Information Sciences Dept., 1700 Main St., Santa Monica, California 90406; J. Vittal & D. A. Henderson, Bolt Beranek & Newman, 50 Moulton St., Cambridge, Massachusetts 02138; and K. Pogran, MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, Massachusetts 02139; The authors' ARPANET addresses are: DCrocker at Rand-Unix, Vittal at BBN-TenexD, Pogran at MIT-Multics, and Henderson at BBN-TenexD.

PREFACE

ARPA's Committee on Computer-Aided Human Communication (CAHCOM) wishes to promulgate a standard for the format of ARPA Network text message (mail) headers which will reasonably meet the needs of the various message service subsystems on the Network today. The authors of this document constitute the CAHCOM subcommittee charged with the task of developing this new standard.

Essentially, we specify a revision to ARPANET Request for Comments (RFC) 561, "Standardizing Network Mail Headers", and RFC 680, "Message Transmission Protocol". This revision removes and compacts portions of the previous syntax and adds several features to network address specification. In particular, we focus on people and not mailboxes as recipients and allow reference to stored address lists. We expect this syntax to provide sufficient capabilities to meet most users' immediate needs and, therefore, give developers enough breathing room to produce a new mail transmission protocol "properly". We believe that there is enough of a consensus in the Network community in favor of such a standard syntax to make possible its adoption at this time. An earlier draft of this specification was published as RFC #724, "Proposed Official Standard for the Format of ARPA Network Messages" and contained extensive discussion of the background and issues in ARPANET mail standards.

This specification was developed over the course of one year, using the ARPANET mail environment, itself, to provide an on-going forum for discussing the capabilities to be included. More than twenty individuals, from across the country, participated in this discussion and we would like to acknowledge their considerable efforts. The syntax of the standard was originally specified in the Backus-Naur Form (BNF) meta-language. Ken L. Harrenstien, of SRI International, was responsible for re-coding the BNF into an augmented BNF which compacts the specification and allows increased comprehensibility.

CONTENTS

PREFACE..... iii

Section
I. INTRODUCTION..... 1

II. FRAMEWORK..... 2

III. SYNTAX..... 4
A. Notational Conventions..... 4
B. Lexical Analysis of Messages..... 5
C. General Syntax of Messages..... 13
D. Syntax of General Addressee Items..... 15
E. Supporting Constructs..... 15

IV. SEMANTICS..... 17
A. Address Fields..... 17
B. Reference Specification Fields..... 22
C. Other Fields and Syntactic Items..... 23
D. Dates and Times..... 24

V. EXAMPLES..... 25
A. Addresses..... 25
B. Address Lists..... 26
C. Originator Items..... 26
D. Complete Headers..... 28

Appendix
A. ALPHABETICAL LISTING OF SYNTAX RULES..... 31
B. SIMPLE PARSING..... 35

REFERENCES..... 37

I. INTRODUCTION

This standard specifies a syntax for text messages which are passed between computer users within the framework of "electronic mail". The standard supersedes the informal standards specified in ARPANET Request for Comments numbers 561, "Standardizing Network Mail Headers", and 680, "Message Transmission Protocol". In this document, a general framework is first described; the formal syntax is then specified, followed by a discussion of the semantics. Finally, a number of examples are given.

This specification is intended strictly as a definition of what is to be passed between hosts on the ARPANET. It is NOT intended to dictate either features which systems on the Network are expected to support, or user interfaces to message creating or reading programs.

A distinction should be made between what the specification REQUIRES and what it ALLOWS. Messages can be made complex and rich with formally-structured components of information or can be kept small and simple, with a minimum of such information. Also, the standard simplifies the interpretation of differing visual formats in messages. These simplifications facilitate the formal specification and indicate what the OFFICIAL semantics are for messages. Only the visual aspect of a message is affected and not the interpretation of information within it. Implementors may choose to retain such visual distinctions.

II. FRAMEWORK

Since there are many message systems which exist outside the ARPANET environment, as well as those within it, it may be useful to consider the general framework, and resulting capabilities and limitations, provided by this standard.

Messages are expected to consist of lines of text. No special provisions are made, at this time, for encoding drawings, facsimile, speech, or structured text.

No significant consideration has been given to questions of data compression or transmission/storage efficiency. The standard, in fact, tends to be very free with the number of bits consumed. For example, field names are specified as free text, rather than special terse codes.

A general "memo" framework is used. That is, a message consists of some information, in a rigid format, followed by the main part of the message, which is text and whose format is not specified in this document. The syntax of several fields of the rigidly-formated ("header") section is defined in this specification; some of the header fields must be included in all messages. The syntax which distinguishes between headers is specified separately from the internal syntax for particular headers. This separation is intended to allow extremely simple parsers to operate on the overall structure of messages, without concern for the detailed structure of individual headers. Appendix B is provided to facilitate construction of these simple parsers. In addition to the fields specified in this document, it is expected that other fields will gain common use. User-defined header fields allow systems to extend their functionality while maintaining a uniform framework. The approach is similar to that of the TELNET protocol, in that a basic standard is defined which includes a mechanism for (optionally) extending itself. As necessary, the authors of this document will regulate the publishing of specifications for these "extension-fields", through the same mechanisms used to publish this document.

Such a framework severely constrains document tone and appearance and is primarily useful for most intra-organization communications and relatively structured inter-organization communication. A more robust environment might allow for multi-font, multi-color, multi-dimension encoding of information. A less robust environment, as is present in most single-machine message systems, would more severely constrain the ability to add fields and the decision to include specific fields. In contrast to paper-based communication, it is interesting to note that the

RECEIVER of a message can exercise an extraordinary amount of control over the message's appearance. The amount of actual control available to message receivers is contingent upon the capabilities of their individual message systems.

III. SYNTAX

This syntax is given in five parts. The first part describes the notation used in the specification. The second part describes the base-level lexical analyzers which feed the higher-level parser described in the succeeding sections. The third part gives a general syntax for messages and standard header fields; and the fourth part specifies the syntax of addresses. A final part specifies some general syntax which supports the other sections.

A. NOTATIONAL CONVENTIONS

These specifications are made in an augmented Backus-Naur Form (BNF). Differences from standard BNF involve the naming of rules, the indication of repetition and of "local" alternatives.

1. Rule naming

Angle brackets ("<", ">") are not used, in general. The name of a rule is simply the name itself, rather than "<name>". Quotation-marks enclose literal text (which may be upper and/or lower case). Certain basic rules are in uppercase, such as SPACE, TAB, CRLF, DIGIT, ALPHA, etc. Angle brackets are used in rule definitions, and in the rest of this document, whenever their presence will facilitate discerning the use of rule names.

2. Parentheses: Local alternatives

Elements enclosed in parentheses are treated as a single element. Thus, "(elem (foo / bar) elem)" allows "(elem foo elem)" and "(elem bar elem)".

3. * construct: Repetition

The character "*" preceding an element indicates repetition. The full form is:

<l>*<m>element

indicating at least <l> and at most <m> occurrences of element. Default values are 0 and infinity so that "*(element)" allows any number, including zero; "1*element" requires at least one; and "1*2element" allows one or two.

4. <number>element

"<n>(element)" is equivalent to "<n>*<n>(element)"; that is, exactly <n> occurrences of (element). Thus 2DIGIT is a 2-digit number, and 3ALPHA is a string of three alphabetic characters.

5. # construct: Lists

A construct "#" is defined, similar to "*", as follows:

<l>#<m>element

indicating at least <l> and at most <m> elements, each separated by one or more commas (","). This makes the usual form of lists very easy; a rule such as '(element *("," element))' can be shown as "1#element". Wherever this construct is used, null elements are allowed, but do not contribute to the count of elements present. That is, "(element),,(element)" is permitted, but counts as only two elements. Therefore, where at least one element is required, at least one non-null element must be present.

6. [optional]

Square brackets enclose optional elements; "[foo bar]" is equivalent to "*1(foo bar)".

7. ; Comments

A semi-colon, set off some distance to the right of rule text, starts a comment which continues to the end of line. This is a simple way of including useful notes in parallel with the specifications.

B. LEXICAL ANALYSIS OF MESSAGES

1. General Description

A message consists of headers and, optionally, a body (i.e. a series of text lines). The text part is just a sequence of lines containing ASCII characters; it is separated from the headers by a null line (i.e., a line with nothing preceding the CRLF).

a. Folding and unfolding of headers

Each header item can be viewed as a single, logical line of ASCII characters. For convenience, the field-body portion of this conceptual entity can be split into a multiple-line representation (i.e., "folded"). The general rule is that wherever there can be linear-white-space (NOT simply LWSP-chars), a CRLF immediately followed by AT LEAST one LWSP-char can instead be inserted. (However, a header's name and the following colon (":"), which occur at the beginning of the header item, may NOT be folded onto multiple lines.) Thus, the single line

```
To:  "Joe Dokes & J. Harvey" <ddd at Host>, JJV at BBN
```

can be represented as

```
To:  "Joe Dokes & J. Harvey" <ddd at Host>,
     JJV at BBN
```

and

```
To:  "Joe Dokes
 & J. Harvey" <ddd at Host>, JJV at BBN
```

The process of moving from this folded multiple-line representation of a header field to its single line representation will be called "unfolding". Unfolding is accomplished by regarding CRLF immediately followed by a LWSP-char as equivalent to the LWSP-char.

b. Structure of header fields

Once header fields have been unfolded, they may be viewed as being composed of a field-name followed by a colon (":"), followed by a field-body. The field-name must be composed of printable ASCII characters (i.e., characters which have values between 33. and 126., decimal, except colon) and LWSP-chars. The field-body may be composed of any ASCII characters (other than an unquoted CRLF, which has been removed by unfolding).

Certain field-bodies of header fields may be interpreted according to an internal syntax which some systems may wish to parse. These fields will be referred to as "structured" fields. Examples include fields containing dates and

addresses. Other fields, such as "Subject" and "Comments", are regarded simply as strings of text.

NOTE: Field-names, unstructured field bodies and structured field bodies each are scanned by their own, INDEPENDENT "lexical" analyzer.

c. Field-names

To aid in the creation and reading of field-names, the free insertion of LWSP-chars is allowed in reasonable places.

Rather than obscuring the syntax specification for field-name with the explicit syntax for these LWSP-chars, the existence of a "lexical" analyzer is assumed. The analyzer interprets the text which comprises the field-name as a sequence of field-name atoms (fnatoms) separated by LWSP-chars

Note that ONLY LWSP-chars may occur between the fnatoms of a field-name and that CRLFs may NOT. In addition, comments are NOT lexically recognized, as such, but parenthesized strings are legal as part of field-names. These constraints are different from what is permissible within structured field bodies. In particular, this means that header field-names must wholly occur on the FIRST line of a folded header item and may NOT be split across two or more lines.

d. Unstructured field bodies

For some fields, such as "Subject" and "Comments", no structuring is assumed; and they are treated simply as texts, like those in the message body. Rules of folding apply to these fields, so that such field bodies which occupy several lines must therefore have the second and successive lines indented by at least one LWSP-char.

e. Structured field bodies

To aid in the creation and reading of structured fields, the free insertion of linear-white-space (which permits folding by inclusion of CRLFs) is allowed in reasonable places. Rather than obscuring the syntax specifications for these structured fields with explicit syntax for this linear-white-space, the existence of another "lexical" analyzer is assumed. This analyzer does not apply for field bodies which are simply unstructured strings of text, as described above. It provides an interpretation of the unfolded text comprising the body of the field as a sequence of lexical symbols. These symbols are:

- individual special characters
- quoted-strings

- comments
- atoms

The first three of these symbols are self-delimiting. Atoms are not; they therefore are delimited by the self-delimiting symbols and by linear-white-space. For the purposes of re-generating sequences of atoms and quoted-strings, exactly one SPACE is assumed to exist and should be used between them. (Also, in Section III.B.3.a, note the rules concerning treatment of multiple continguous LWSP-chars.)

So, for example, the folded body of an address field

```
":sysmail"@  Some-Host,
Muhammed(I am  the greatest)Ali  at(the)WBA
```

is analyzed into the following lexical symbols and types:

":sysmail"	quoted string
@	special
Some-Host	atom
,	special
Muhammed	atom
(I am the greatest)	comment
Ali	atom
at	atom
(the)	comment
WBA	atom

The cononical representations for the data in these addresses are the following strings (note that there is exactly one SPACE between words):

:sysmail at Some-Host

and

Muhammed Ali at WBA

2. Formal Definitions

The first four rules, below, indicate a meta-syntax for fields, without regard to their particular type or internal syntax. The remaining rules define basic syntactic structures which are used by the rules in Sections III.C, III.D, and III.E.

```
field      =  field-name ":" [ field-body ] CRLF

field-name =  fnatom *( LWSP-char [fnatom] )
```

```
fnatom      =  1*<any CHAR, excluding CTLs, SPACE, and ":">

field-body  =  field-body-contents
               [CRLF LWSP-char field-body]

field-body-contents = <the TELNET ASCII characters making up the
                       field-body, as defined in the following sections,
                       and consisting of combinations of atom, quoted-
                       string, and specials tokens, or else consisting of
                       texts>

                                                  ; (  Octal, Decimal.)
CHAR        =  <any TELNET ASCII character>       ; (  0-177,  0.-127.)
ALPHA       =  <any TELNET ASCII alphabetic character>
                                                  ; (101-132, 65.- 90.)
                                                  ; (141-172, 97.-122.)
DIGIT       =  <any TELNET ASCII digit>           ; ( 60- 71, 48.- 57.)
CTL         =  <any TELNET ASCII control          ; (  0- 37,  0.- 31.)
                character and DEL>                ; (    177,    127.)
CR          =  <TELNET ASCII carriage return>     ; (     15,     13.)
LF          =  <TELNET ASCII linefeed>            ; (     12,     10.)
SPACE       =  <TELNET ASCII space>               ; (     40,     32.)
HTAB        =  <TELNET ASCII horizontal-tab>      ; (     11,      9.)
<">         =  <TELNET ASCII quote mark>          ; (     42,     34.)
CRLF        =  CR LF

LWSP-char   =  SPACE / HTAB                       ; semantics = SPACE
linear-white-space =  1*([CRLF] LWSP-char)        ; semantics = SPACE
                                                  ; CRLF => folding

specials    =  "(" / ")" / "<" / ">" / "@"        ; To use in a word,
            /  "," / ";" / ":" / "\" / <">        ;  word must be a
                                                  ;  quoted-string.

delimiters  =  specials / comment / linear-white-space

text        =  <any CHAR, including bare          ; => atoms, specials,
                CR and/or bare LF, but NOT        ;  comments and
                including CRLF>                   ;  quoted-strings are
                                                  ;  NOT interpreted.

atom        =  1*<any CHAR except specials and CTLs>

quoted-string = <"> *(qtext/quoted-pair) <">; Any number of qtext
                                                  ;   chars or any
                                                  ;   quoted char.

qtext       =  <any CHAR excepting <">            ; => may be folded
                and CR, and including
                linear-white-space>
```

```
comment     =  "(" *(ctext / comment / quoted-pair) ")"
ctext       =  <any CHAR excluding "(",     ; => may be folded
                ")" and CR, and including
                linear-white-space>

quoted-pair =  "\" CHAR
```

3. Clarifications

a. "White space"

Remember that in field-names and structured field bodies, MULTIPLE LINEAR WHITE SPACE TELNET ASCII CHARACTERS (namely HTABs and SPACEs) ARE TREATED AS SINGLE SPACES AND MAY FREELY SURROUND ANY SYMBOL. In all header fields, the only place in which at least one space is REQUIRED is at the beginning of continuation lines in a folded field. When passing text to processes which do not interpret text according to this standard (e.g., ARPANET FTP mail servers), then exactly one SPACE should be used in place of arbitrary linear-white-space and comment sequences.

WHEREVER A MEMBER OF THE LIST OF <DELIMITER>S IS ALLOWED, LWSP-CHARS MAY ALSO OCCUR BEFORE AND/OR AFTER IT.

Writers of mail-sending (i.e. header generating) programs should realize that there is no Network-wide definition of the effect of horizontal-tab TELNET ASCII characters on the appearance of text at another Network host; therefore, the use of tabs in message headers, though permitted, is discouraged.

Note that during transmissions across the ARPANET using TELNET NVT connections, data must conform to TELNET NVT conventions (e.g., CR must be followed by either LF, making a CRLF, or <null>, if the CR is to stand alone).

b. Comments

Comments are detected as such only within field-bodies of structured fields. A comment is a set of TELNET ASCII characters, which is not within a quoted-string and which is enclosed in matching parentheses; parentheses nest, so that if an unquoted left parenthesis occurs in a comment string, there must also be a matching right parenthesis. When a comment is used to act as the delimiter between a sequence of two lexical symbols, such as two atoms, it is lexically equivalent with one SPACE, for the purposes of regenerating the sequence, such as when passing the sequence onto an FTP mail server.

In particular comments are NOT passed to the FTP server, as part of a MAIL or MLFL command, since comments are not part of the "formal" address.

If a comment is to be "folded" onto multiple lines, then the syntax for folding must be adhered to. (See items III.B.1.a, above, and III.B.3.f, below.) Note that the official semantics therefore do not "see" any unquoted CRLFs which are in comments, although particular parsing programs may wish to note their presence. For these programs, it would be reasonable to interpret a "CRLF LWSP-char" as being a CRLF which is part of the comment; i.e., the CRLF is kept and the LWSP-char is discarded. Quoted CRLFs (i.e., a backslash followed by a CR followed by a LF) still must be followed by at least one LWSP-char.

c. Delimiting and quoting characters

The quote character (backslash) and characters which delimit syntactic units are not, generally, to be taken as data which are part of the delimited or quoted unit(s). The one exception is SPACE. In particular, the quotation-marks which define a quoted-string, the parentheses which define a comment and the backslash which quotes a following character are NOT part of the quoted-string, comment or quoted character. A quotation-mark which is to be part of a quoted-string, a parenthesis which is to be part of a comment and a backslash which is to be part of either must each be preceded by the quote-character backslash ("\"). Note that the syntax allows any character to be quoted within a quoted-string or comment; however only certain characters MUST be quoted to be included as data. These characters are those which are not part of the alternate text group (i.e., ctext or qtext).

A single SPACE is assumed to exist between contiguous words in a phrase, and this interpretation is independent of the actual number of LWSP-chars which the creator places between the words. To include more than one SPACE, the creator must make the LWSP-chars be part of a quoted-string.

Quotation marks which delimit a quoted string and backslashes which quote the following character should NOT accompany the quoted-string when the string is used with processes that do not interpret data according to this specification (e.g., ARPANET FTP mail servers).

d. Quoted-strings

Where permitted (i.e., in words in structured fields) quoted-strings are treated as a single symbol (i.e. equivalent to an atom, syntactically). If a quoted-string is to be "folded" onto multiple lines, then the syntax for folding must be adhered to. (See items III.B.1.a, above, and III.B.3.f, below.) Note that the official semantics therefore do not "see" any bare CRLFs which are in quoted-strings, although particular parsing programs may wish to note their presence. For these programs, it would be reasonable to interpret a "CRLF LWSP-char" as being a CRLF which is part of the quoted-string; i.e., the CRLF is kept and the LWSP-char is discarded. Quoted CRLFs (i.e., a backslash followed by a CR followed by a LF) are also subject to rules of folding, but the presence of the quoting character (backslash) explicitly indicates that the CRLF is data to the quoted string. Stripping off the first following LWSP-char is also appropriate when parsing quoted CRLFs.

e. Bracketing characters

There are three types of brackets which must be well nested:

- o Parentheses are used to indicate comments.
- o Angle brackets ("<" and ">") are generally used to indicate the presence of at least one machine-usable code (e.g., delimiting mailboxes).
- o Colon/semi-colon (":" and ";") are used in address specifications to indicate that the included list of addresses are to be treated as a group.

f. Case independence of certain specials atoms

Certain atoms, which are represented in the syntax as literal alphabetic strings, can be represented in any combination of upper and lower case. These are:

- field-name,
- "Include", "Postal" and equivalent atoms in a ":"<atom>":" address specification,
- "at", in a host-indicator,
- node,
- day-of-week,
- month, and
- zones.

When matching an atom against one of these literals, case is to be ignored. For example, the field-names "From", "FROM",

"from", and even "FroM" should all be treated identically. However, the case shown in this specification is suggested for message-creating processes. Note that, at the level of this specification, case IS relevant to other words and texts. Also see Section IV.A.1.f, below.

g.)Folding long lines

Each header item (field of the message) may be represented on exactly one line consisting of the name of the field and its body; this is what the parser sees. For readability, it is recommended that the field-body portion of long header items be "folded" onto multiple lines of the actual header. "Long" is commonly interpreted to mean greater than 65 or 72 characters. The former length is recommended as a limit, but it is not imposed by this standard.

h. Backspace characters

Backspace TELNET ASCII characters (ASCII BS, decimal 8.) may be included in texts and quoted-strings to effect overstriking; however, any use of backspaces which effects an overstrike to the left of the beginning of the text or quoted-string is prohibited.

C. GENERAL SYNTAX OF MESSAGES:

NOTE: Due to an artifact of the notational conventions, the syntax indicates that, when present, "Date", "From", "Sender", and "Reply-To" fields must be in a particular order. These header items must be unique (occur exactly once). However header fields, in fact, are NOT required to occur in any particular order, except that the message body must occur AFTER the headers. For readability and ease of parsing by simple systems, it is recommended that headers be sent in the order "Date", "From", "Subject", "Sender", "To", "cc", etc. This specification permits multiple occurrences of most optional-fields. However, their interpretation is not specified here, and their use is strongly discouraged.

The following syntax for the bodies of various fields should be thought of as describing each field body as a single long string (or line). The section on Lexical Analysis (section II.B) indicates how such long strings can be represented on more than one line in the actual transmitted message.

```
message      =  fields *( CRLF *text )        ; Everything after
                                              ;  first null line
                                              ;  is message body

fields       =  date-field                    ; Creation time-stamp
                originator-fields             ;  & author id are
                *optional-field               ;  required: others
                                              ;  are all optional

originator-fields =
                (  "From"     ":" mailbox     ; Single author
                  ["Reply-To" ":" #address] )
              /  (  "From"     ":" 1#address   ; Multiple authors &
                    "Sender"   ":" mailbox     ;  may have non-mach-
                  ["Reply-To" ":" #address] ); ine addresses

date-field   =  "Date"       ":" date-time

optional-field  =
                "To"         ":" #address
              /  "cc"        ":" #address
              /  "bcc"       ":" #address     ; Blind carbon
              /  "Subject"   ":" *text
              /  "Comments"  ":" *text
              /  "Message-ID" ":" mach-id      ; Only one allowed
              /  "In-Reply-To"":" #(phrase / mach-id)
              /  "References" ":" #(phrase / mach-id)
              /  "Keywords"   ":" #phrase
              /  extension-field               ; To be defined in
                                               ;  supplemental
                                               ;  specifications
              /  user-defined-field            ; Must have unique
                                               ;  field-name & may
                                               ;  be pre-empted

extension-field = <Any field which is defined in a document
                published as a formal extension to this
                specification>

user-defined-field = <Any field which has not been defined in
                this specification or published as an extension to
                this specification; names for such fields must be
                unique and may be preempted by published
                extensions>
```

D. SYNTAX OF GENERAL ADDRESSEE ITEMS

```
address      =  host-phrase                        ; Machine mailbox
             / ( [phrase] "<" #address ">")        ; Individual / List
             / ( [phrase] ":" #address ";")        ; Group
             /  quoted-string                      ; Arbitrary text
             / (":" ( "Include"                    ; File, w/ addr list
                    / "Postal"                     ; (U.S.) Postal addr
                    /  atom )                      ; Extended data type
                ":" address)

mailbox      =  host-phrase /  (phrase mach-id)

mach-id      =  "<" host-phrase ">"                ; Contents must never
                                                   ;  be modified!
```

E. SUPPORTING CONSTRUCTS

```
host-phrase =  phrase  host-indicator              ; Basic address

host-indicator =  1*( ("at" / "@") node )          ; Right-most node is
                                                   ;  at top of network
                                                   ;  hierarchy; left-
                                                   ;  most must be host

node         =  word / 1*DIGIT                     ; Official host or
                                                   ;  network name or
                                                   ;  decimal address


date-time    =  [ day-of-week "," ] date time

day-of-week =  "Monday"    / "Mon"  / "Tuesday"   / "Tue"
             /  "Wednesday" / "Wed"  / "Thursday"  / "Thu"
             /  "Friday"    / "Fri"  / "Saturday"  / "Sat"
             /  "Sunday"    / "Sun"

date         =  1*2DIGIT ["-"] month               ; day month year
                ["-"] (2DIGIT /4DIGIT)             ;  e.g. 20 Aug [19]77

month        =  "January"   / "Jan"  / "February"  / "Feb"
             /  "March"     / "Mar"  / "April"     / "Apr"
             /  "May"                / "June"      / "Jun"
             /  "July"      / "Jul"  / "August"    / "Aug"
             /  "September" / "Sep"  / "October"   / "Oct"
             /  "November"  / "Nov"  / "December"  / "Dec"
```

```
time        =  hour zone                        ; ANSI and Military
                                                ;  (seconds optional)

hour        =  2DIGIT [":"] 2DIGIT [ [":"] 2DIGIT ]
                                                ; 0000[00] - 2359[59]

zone        =  ( ["-"] ( "GMT"                  ; Relative to GMT:
                                                ; North American
                    /  "NST" /                  ;  Newfoundland:-3:30
                    /  "AST" / "ADT"            ;  Atlantic: - 4/ - 3
                    /  "EST" / "EDT"            ;  Eastern:  - 5/ - 4
                    /  "CST" / "CDT"            ;  Central:  - 6/ - 5
                    /  "MST" / "MDT"            ;  Mountain: - 7/ - 6
                    /  "PST" / "PDT"            ;  Pacific:  - 8/ - 7
                    /  "YST" / "YDT"            ;  Yukon:    - 9/ - 8
                    /  "HST" / "HDT"            ;  Haw/Ala   -10/ - 9
                    /  "BST" / "BDT"            ;  Bering:   -11/ -10
                       1ALPHA       ))          ; Military: Z = GMT;
                                                ;  A:-1; (J not used)
                                                ;  M:-12; N:+1; Y:+12
              / ( ("+" / "-") 4DIGIT )           ; Local differential
                                                ;  hours/min. (HHMM)

phrase      =  1*word                           ; Sequence of words.
                                                ;  Separation seman-
                                                ;  tically = SPACE

word        =  atom / quoted-string
```

IV. SEMANTICS

A. ADDRESS FIELDS

1. General

a. The phrase part of a host-phrase in an address specification (i.e., the host's name for the mailbox) is understood to be whatever the receiving FTP Server allows (for example, TENEX systems do not now understand addresses of the form "P. D. Q. Bach", but another system might).

Note that a mailbox is a conceptual entity which does not necessarily pertain to file storage. For example, some sites may choose to print mail on their line printer and deliver the output to the addressee's desk.

An individual may have several mailboxes and a group of individuals may wish to receive mail as a single unit (i.e., a distribution list). The second and third alternatives of an address list (#address) allow naming a collection of subordinate addresses list(s). Recipient mailboxes are specified within the bracketed part ("<" - ">" or ":" - ";") of such named lists. The use of angle-brackets ("<", ">") is intended for the cases of individuals with multiple mailboxes and of special mailbox lists; it is not expected to be nested more than one level, although the specification allows such nesting. The use of colon/semi-colon (":", ";") is intended for the case of groups. Groups can be expected to nest (i.e., to contain subgroups). For both individuals and groups, a copy of the transmitted message is to be sent to EACH mailbox listed. For the case of a special list, treatment of addresses is defined in the relevant subsections of this section.

b. The inclusion of bare quoted-strings as addresses (i.e., the fourth address-form alternative) is allowed as a syntactic convenience, but no semantics are defined for their use. However, it is reasonable, when replicating an address list, to replicate ALL of its members, including quoted-strings.

c. ":Include:" specifications are used to refer to one or more locations containing stored address lists (#address). If more than one location is referenced, the address part of the Include phrase must be a list (#address) surrounded by angle-brackets, as per the "Individual / List" alternative of <address>. Constituent addresses must resolve to a host-

phrase; only they have any meaning within this construct. The phrase part of indicated host-phrases should contain text which the referenced host can resolve to a file. This standard is not a protocol and so does not prescribe HOW data is to be retrieved from the file. However, the following requirements are made:

- The file must be accessible through the local operating system interface (if it exists), given adequate user access rights; and

- If a host has an FTP server and a user is able to retrieve any files from the host using that server, then the file must be accessible through FTP, using DEFAULT transfer settings, given adequate user access rights.

It is intended that this mechanism allow programs to retrieve such lists automatically.

The interpretation of such a file reference follows. This is not intended to imply any particular implementation scheme, but is presented to aid in understanding the notion of including file contents in address lists:

- Elements of the address list part are alternates and the contents of ONLY ONE of them are to be included in the resultant address list.

- The contents of the file indicated by a member host-phrase are treated as an address list and are inserted as an address list (#address) in the position of the path item in the syntax. That is, the TELNET ASCII characters specifying the entire Include <address> is replaced by the contents of one of the files to which the host-phrase(s), of the address list (#address), refers. Therefore, the contents of each file, indicated by an Include address, must be syntactically self-contained and must adhere to the full syntax prescribed herein for an address list.

d. ":Postal:" specifications are used to indicate (U.S.) postal addresses, but can be treated the same as quoted-string addresses. To reference a list of postal addresses, the list must conform to the "Individual / List" alternative of <address>. The ":Include:" alternative also is valid.

e. The "':' atom ':'" syntax is intended as a general mechanism for indicating specially data-typed addresses. As with extension-fields, the authors of this document will regulate

the publishing of specifications for these extended data-types. In the absence of defined semantics, any occurrence of an address in this form may be treated as a quoted-string address.

f. A node name must be THE official name of a network or a host, or else a decimal number indicating the Network address for that network or host, at the time the message is created. The USE OF NUMBERS IS STRONGLY DISCOURAGED and is permitted only due to the occasional necessity of bypassing local name tables. For the ARPANET, official names are maintained by the Network Information Center at SRI International, Menlo Park, California.

Whenever a message might be transmitted or migrate to a host on another network, full hierarchical addresses must be specified. These are indicated as a series of words, separated by at-sign or "at" indications. The communication environment is assumed to consist of a collection of networks organized as independent "trees" except for connections between the root nodes. That is, only the roots can act as gateways between these independent networks. While other actual connections may exist, it is believed that presuming this type of organization will provide a reliable method for describing VALID, if not EFFICIENT, paths between hosts. A typical full mailbox specification might therefore look like:

Friendly User @ hosta @ local-net1 @ major-netq

In the simplest case, a mail-sending host should transmit the message to the node which is mentioned last (farthest to the right), strip off that node reference from the specification, and then pass the remaining host-phrase to the recipient host (in the ARPANET, its FTP server) for it to process. This treats the remaining portion of the host-indicator merely as the terminating part of the phrase.

NOTE: When passing any portion of a host-indicator onto a process which does not interpret data according to this standard (e.g., ARPANET FTP servers), "@" must be used and not "at" and it must not be preceded or followed by any LWSP-chars. Using the above example, the following string would be passed to the major-netq gateway:

Friendly User@hosta@localnet-1

When the sending host has more knowledge of the network environment, then it should send the message along a more efficient path, making appropriate changes to the form of the host-phrase which it gives to the recipient host.

To use the above specification as an example: If a sending hostb also were part of local-net1, then it could send the message directly to hosta and would give only the phrase "Friendly User" to hosta's mail-receiving program. If hostb were part of local-net2, along with hostc, and happened to know that hosta and hostc were part of another local-net, then hostb could send the message to hostc to the address "Friendly User@hosta".

The phrase in a host-phrase is intended to be meaningful only to the indicated receiving host. To all other hosts, the phrase is to be treated as an uninterpreted string. No case transformations should be (automatically) performed on the phrase. The phrase is passed to the local host's mail sending program; it is the responsibility of the destination host's mail receiving (distribution) program to perform case mapping on this phrase, if required, to deliver the mail.

2. Originator Fields

WARNING: The standard allows only a subset of the combinations possible with the From, Sender, and Reply-To fields. The limitation is intentional.

a. From

This field contains the identity of the person(s) who wished this message to be sent. The message-creation process should default this field to be a single machine address, indicating the AGENT (person or process) entering the message. If this is NOT done, the "Sender" field MUST be present; if this IS done, the "Sender" field is optional.

b. Sender

This field contains the identity of the AGENT (person or process) who sends the message. It is intended for use when the sender is not the author of the message, or to indicate who among a group of authors actually sent the message. If the contents of the "Sender" field would be completely redundant with the "From" field, then the "Sender" field need not be present and its use is discouraged (though still legal); in particular, the "Sender" field MUST be present if it is NOT the same as the "From" Field.

The Sender host-phrase includes a phrase which must correspond to a specific agent (i.e., a human user or a computer program) rather than a standard address. This indicates the expectation that the field will identify the single AGENT (person or process) responsible for sending the

mail and not simply include the name of a mailbox from which the mail was sent. For example in the case of a shared login name, the name, by itself, would not be adequate. The phrase part of the host-phrase, which refers to this agent, is expected to be a computer system term, and not (for example) a generalized person reference which can be used outside the network text message context.

Since the critical function served by the "Sender" field is the identification of the agent responsible for sending mail and since computer programs cannot be held accountable for their behavior, is strongly recommended that when a computer program generates a message, the HUMAN who is responsible for that program be referenced as part of the "Sender" field host-phrase.

c. Reply-To

This field provides a general mechanism for indicating any mailbox(es) to which responses are to be sent. Three typical uses for this feature can be distinguished. In the first case, the author(s) may not have regular machine-based mailboxes and therefore wish(es) to indicate an alternate machine address. In the second case, an author may wish additional persons to be made aware of, or responsible for, responses; responders should send their replies to the "Reply-To" mailbox(es) listed in the original message. A somewhat different use may be of some help to "text message teleconferencing" groups equipped with automatic distribution services: include the address of that service in the "Reply-To" field of all messages submitted to the teleconference; then participants can "reply" to conference submissions to guarantee the correct distribution of any submission of their own.

Reply-To fields are NOT required to contain any machine addresses (i.e., host-phrases). Note, however, that the absence of even one valid network address will tend to prevent software systems from automatically assisting users in conveniently responding to mail.

NOTE: For systems which automatically generate address lists for replies to messages, the following recommendations are made:

- The receiver, when replying to a message, should NEVER automatically include the "Sender" host-phrase in the reply's address list;

- If the "Reply-To" field exists, then the reply should go ONLY to the addresses indicated in that field and not to the addresses indicated in the "From" field.

(Extensive examples are provided in Section V.) This recommendation is intended only for originator-fields and is not intended to suggest that replies should not also be sent to the other recipients of this message. It is up to the respective mail handling programs to decide what additional facilities will be provided.

3. Receiver Fields

a. To

This field contains the identity of the primary recipients of the message.

b. cc

This field contains the identity of the secondary recipients of the message.

b. Bcc

This field contains the identity of additional recipients of the message. The contents of this field are not included in copies of the message sent to the primary and secondary recipients. Some systems may choose to include the text of the "Bcc" field only in the author(s)'s copy, while others may also include it in the text sent to all those indicated in the "Bcc" list.

B. REFERENCE SPECIFICATION FIELDS

1. Message-ID

This field contains a unique identifier (the phrase) which refers to THIS version of THIS message. The uniqueness of the message identifier is guaranteed by the host which generates it. This identifier is intended to be machine readable and not necessarily meaningful to humans. A message identifier pertains to exactly one instantiation of a particular message; subsequent revisions to the message should each receive a new message identifier.

2. In-Reply-To

The contents of this field identify previous correspondence which this message answers. Note that if message identifiers are used in this field, they must use the mach-id specification format.

3. References

The contents of this field identify other correspondence which this message references. Note that if message identifiers are used, they must use the mach-id specification format.

4. Keywords

This field contains keywords or phrases, separated by commas.

C. OTHER FIELDS AND SYNTACTIC ITEMS

1. Subject

The "Subject" field is intended to provide as much information as necessary to adequately summarize or indicate the nature of the message.

2. Comments

Permits adding text comments onto the message without disturbing the contents of the message's body.

3. Extension-field

A relatively limited number of common fields have been defined in this document. As network mail requirements dictate, additional fields may be standardized. The authors of this document will regulate the publishing of such definitions as extensions to the basic specification.

4. User-defined-field

Individual users of network mail are free to define and use additional header fields. Such fields must have names which are not already used in the current specification or in any definitions of extension-fields, and the overall syntax of these user-defined-fields must conform to this specification's rules for delimiting and folding fields. Due to the extension-field publishing process, the name of a user-defined-field may be pre-empted.

D. DATES AND TIMES

If included, day-of-week must be the day implied by the date specification.

Time zone may be indicated in several ways. The military standard uses a single character for each zone. "Z" is Greenwhich Mean Time; "A" indicates one hour earlier, and "M" indicates 12 hours earlier; "N" is one hour later, and "Y" is 12 hours later. The letter "J" is not used. The other remaining two forms are taken from ANSI standard X3.51-1975. One allows explicit indication of the amount of offset from GMT; the other uses common 3-character strings for indicating time zones in North America.

V. EXAMPLES

A. ADDRESSES

1. Alfred E. Neuman <Neuman at BBN-TENEXA>

2. Neuman@BBN-TENEXA

These two "Alfred E. Neuman" examples have identical semantics, as far as the operation of the local host's mail sending (distribution) program (also sometimes called its "mailer") and the remote host's FTP server are concerned. In the first example, the "Alfred E. Neuman" is ignored by the mailer, as "Neuman at BBN-TENEXA" completely specifies the recipient. The second example contains no superfluous information, and, again, "Neuman@BBN-TENEXA" is the intended recipient.

3. Al Neuman at BBN-TENEXA

This is identical to "Al Neuman <Al Neuman at BBN-TENEXA>". That is, the full phrase, "Al Neuman", is passed to the FTP server. Note that not all FTP servers accept multi-word identifiers; and some that do accept them will treat each word as a different addressee (in this case, attempting to send a copy of the message to "Al" and a copy to "Neuman").

4. "George Lovell, Ted Hackle" <Shared-Mailbox at Office-1>

This form might be used to indicate that a single mailbox is shared by several users. The quoted string is ignored by the originating host's mailer, as "Shared-Mailbox at Office-1" completely specifies the destination mailbox.

4. Wilt (the Stilt) Chamberlain at NBA

The "(the Stilt)" is a comment, which is NOT included in the destination mailbox address handed to the originating system's mailer. The address is the string "Wilt Chamberlain", with exactly one space between the first and second words. (The quotation marks are not included.)

B. ADDRESS LISTS

```
Gourmets:  Pompous Person <WhoZiWhatZit at Cordon-Bleu>,
           Cooks:  Childs at WGBH, Galloping Gourmet at
                   ANT (Australian National Television);,
           Wine Lovers:  Cheapie at Discount-Liquors,
                         Port at Portugal;;,
Jones at SEA
```

This group list example points out the use of comments, the nesting of groups, and the mixing of addresses and groups. Note that the two consecutive semi-colons preceding "Jones at SEA" mean that Jones is NOT a member of the Gourmets group.

C. ORIGINATOR ITEMS

1. Author-sent

George Jones logs into his Host as "Jones". He sends mail himself.

```
From:  Jones at Host
```

or

```
From:  George Jones <Jones at Host>
```

2. Secretary-sent

George Jones logs in as Jones on his Host. His secretary, who logs in as Secy on Shost sends mail for him. Replies to the mail should go to George, of course.

```
From:    George Jones <Jones at Host>
Sender:  Secy at SHost
```

3. Shared directory or unrepresentative directory-name

George Jones logs in as Group at Host. He sends mail himself; replies should go to the Group mailbox.

```
From:  George Jones <Group at Host>
```

4. Secretary-sent, for user of shared directory

George Jones' secretary sends mail for George in his capacity as a member of Group while logged in as Secy at Host. Replies should go to Group.

```
From:   George Jones<Group at Host>
Sender: Secy at Host
```

Note that there need not be a space between "Jones" and the "<", but adding a space enhances readability (as is the case in other examples).

5. Secretary acting as full agent of author

George Jones asks his secretary (Secy at Host) to send a message for him in his capacity as Group. He wants his secretary to handle all replies.

```
From:     George Jones <Group at Host>
Sender:   Secy at Host
Reply-To: Secy at Host
```

6. Agent for user without online mailbox

A non-ARPANET user friend of George's, Sarah, is visting. George's secretary sends some mail to a friend of Sarah in computer-land. Replies should go to George, whose mailbox is Jones at Host.

```
From:     Sarah Friendly
Sender:   Secy at Host
Reply-To: Jones at Host
```

7. Sent by member of a committee

George is a member of a committee. He wishes to have any replies to his message go to all committee members.

```
From:     George Jones
Sender:   Jones at Host
Reply-To: Big-committee: Jones at Host,
                         Smith at Other-Host,
                         Doe at Somewhere-Else;
```

Note that if George had not included himself in the enumeration of Big-committee, he would not have gotten an implicit reply; the presence of the "Reply-to" field SUPERSEDES the sending of a reply to the person named in the "From" field.

8. Example of INCORRECT use

George desires a reply to go to his secretary; therefore his secretary leaves his mailbox address off the "From" field, leaving only his name, which is not, itself, a mailbox address.

```
From:   George Jones
Sender: Secy at SHost
```

THIS IS NOT PERMITTED. Replies are NEVER implicitly sent to the "Sender"; George's secretary should have used the "Reply-To" field, or the mail creating program should have forced the secretary to.

9. Agent for member of a committee

George's secretary sends out a message which was authored jointly by all the members of the "Big-committee".

```
From:   Big-committee: Jones at Host,
                       Smith at Other-Host,
                       Doe at Somewhere-Else;
Sender: Secy at SHost
```

D. COMPLETE HEADERS

1. Minimum required:

```
Date:  26 August 1976 1429-EDT
From:  Jones at Host
```

2. Using some of the additional fields:

```
Date: 26 August 1976 1430-EDT
From:George Jones<Group at Host>
Sender:Secy at SHOST
To:Al Neuman at Mad-Host,
          Sam Irving at Other-Host
Message-ID:  <some string at SHOST>
```

3. About as complex as you're going to get:

```
Date       : 27 Aug 1976 0932-PDT
From       : Ken Davis <KDavis at Other-Host>
Subject    : Re: The Syntax in the RFC
Sender     : KSecy at Other-Host
Reply-To   : Sam Irving at Other-Host
To         : George Jones <Group at Host>,
             Al Neuman at Mad-Host
cc         : Important folk:
             Tom Softwood <Balsa at Another-Host>,
             Sam Irving at Other-Host;,
             Standard Distribution::Include:
              </main/davis/people/standard at Other-Host,
               "<Jones>standard.dist.3" at Tops-20-Host>,
             (The following Included Postal list is part
             of Standard Distribution.)
             :Postal::Include: Non-net-addrs@Other-host;,
             :Postal: "Sam Irving, P.O. Box 001, Las Vegas,
                      Nevada"  (So that he can stay
                      apprised of the situation)
Comment    : Sam is away on business. He asked me to handle
             his mail for him.  He'll be able to provide  a
             more accurate explanation when he returns
             next week.
In-Reply-To: <some string at SHOST>
Special (action):  This is a sample of multi-word field-
             names, using a range of characters.  There
             could also be a field-name "Special (info)".
Message-ID: <4231.629.XYzi-What at Other-Host>
```

APPENDIX

A. ALPHABETICAL LISTING OF SYNTAX RULES

```
address     =  host-phrase / quoted-string
            / (*phrase "<" #address ">" )
            / (*phrase ":" #address ";" )
            / (":" ("Include" / "Postal" / atom) ":" address)
ALPHA       =  <any TELNET ASCII alphabetic character>
atom        =  1*<any CHAR except specials and CTLs>

CHAR        =  <any TELNET ASCII character>
comment     =  "(" *(ctext / comment / quoted-pair) ")"
CR          =  <TELNET ASCII carriage return>
CRLF        =  CR LF
ctext       =  <any CHAR excluding "(", ")", CR, LF and
               including linear-white-space>
CTL         =  <any TELNET ASCII control character and DEL>

date        =  1*2DIGIT ["-"] month ["-"] (2DIGIT /4DIGIT)
date-field  =  "Date"       ":" date-time
date-time   =  [ day-of-week "," ] date time
day-of-week =  "Monday"    / "Mon"  / "Tuesday"   / "Tue"
            /  "Wednesday" / "Wed"  / "Thursday"  / "Thu"
            /  "Friday"    / "Fri"  / "Saturday"  / "Sat"
            /  "Sunday"    / "Sun"
delimiters  =  specials / comment / linear-white-space
DIGIT       =  <any TELNET ASCII digit>

extension-field = <Any field which is defined in a document
                  published as a formal extension to this
                  specification>

field       =  field-name ":" [ field-body ] CRLF

fields      =  date-field  originator-fields  *optional-field
field-body  =  field-body-contents
               [CRLF LWSP-char field-body]
field-body-contents = <the TELNET ASCII characters making up the
                  field-body, as defined in the following sections,
                  and consisting of combinations of atom, quoted-
                  string, and specials tokens, or else consisting of
                  texts>
field-name  =  fnatom *(LWSP-char [fnatom])
fnatom      =  1*<any CHAR, excluding CTLs, SPACE, and ":">
```

```
host-indicator =  1*( ("at" / "@") node )
host-phrase =  phrase  host-indicator
hour        =  2DIGIT [":"] 2DIGIT [ [":"] 2DIGIT ]
HTAB        =  <TELNET ASCII horizontal-tab>

LF          =  <TELNET ASCII linefeed>
linear-white-space =  1*([CRLF] LWSP-char)
LWSP-char   = SPACE / HTAB

mach-id     =  "<" host-phrase ">"
mailbox     =  host-phrase /  (phrase mach-id)
message     =  fields *(CRLF *text)
month       =  "January"   / "Jan"  / "February"  / "Feb"
            /  "March"     / "Mar"  / "April"     / "Apr"
            /  "May"                / "June"      / "Jun"
            /  "July"      / "Jul"  / "August"    / "Aug"
            /  "September" / "Sep"  / "October"   / "Oct"
            /  "November"  / "Nov"  / "December"  / "Dec"

node        =  word / 1*DIGIT

optional-field  =
               "To"          ":" #address
            /  "cc"          ":" #address
            /  "bcc"         ":" #address
            /  "Subject"     ":" *text
            /  "Comments"    ":" *text
            /  "Message-ID" ":" mach-id
            /  "In-Reply-To"":" #(phrase / mach-id)
            /  "References" ":" #(phrase / mach-id)
            /  "Keywords"   ":" #phrase
            /  extension-field
            /  user-defined-field
originator-fields =
               (  "From"     ":" mailbox
                 ["Reply-To" ":" #address] )
            /  (  "From"     ":" 1#address
                  "Sender"   ":" mailbox
                 ["Reply-To" ":" #address] )

phrase      =  1*word

quoted-pair =  "\" CHAR
quoted-string =  <">  *(qtext / quoted-pair)  <">
qtext       =  <any CHAR except <">, CR, or LF and including
               linear-white-space>
SPACE       =  <TELNET ASCII space>
specials    =  "(" / ")" / "<" / ">" / "@"/ "," / ";" / ":"
            /  "\" / <">

text        =  <any CHAR, including bare CR and/or bare LF, but
               NOT including CRLF>
```

Appendix
A. Alphabetical Listing of Syntax Rules

```
time            =  hour zone

user-defined-field = <Any field which has not been defined in
                      this specification or published as an extension to
                      this specification; names for such fields must be
                      unique and may be preempted by putlished
                      extensions>

word            =  atom / quoted-string

zone            =  ( ("+" / "-") 4DIGIT )
                   / ( ["-"] (1ALPHA
                     / "GMT" / "NST"  / "AST" / "ADT" / "EST" / "EDT"
                     / "CST" / "CDT"  / "MST" / "MDT" / "PST" / "PDT"
                     / "YST" / "YDT"  / "HST" / "HDT" / "BST" / "BDT" ))

<">             =  <TELNET ASCII quote mark>
```

B. SIMPLE PARSING

Some mail-reading software systems may wish to perform only minimal processing, ignoring the internal syntax of structured field-bodies and treating them the same as unstructured-field-bodies. Such software will need only to distinguish:

- Header fields from the message body,
- Beginnings of fields from lines which continue fields,
- Field-names from field-contents.

The abbreviated set of syntactic rules which follows will suffice for this purpose. They describe a limited view of messages and are a subset of the syntactic rules provided in the main part of this specification. One small exception is that the contents of field-bodies consist only of text:

SYNTAX:

message = *field *(CRLF *text)

field = field-name ":" [field-body] CRLF

field-name = fnatom *(LWSP-char [fnatom])

fnatom = 1*<any CHAR, excluding CTLs, SPACE, and ":">

field-body = *text [CRLF LWSP-char field-body]

SEMANTICS:

Headers occur before the message body and are terminated by a null line (i.e., two contiguous CRLFs).

A line which continues a header field begins with a SPACE or HTAB character, while a line beginning a field starts with a printable character which is not a colon.

A field-name consists of one or more printable characters (excluding colon), each separated by one or more SPACES or HTABS. A field-name MUST be contained on one line. Upper and lower case are not distinguished when comparing field-names.

REFERENCES

ANSI. Representations of universal time, local time differentials, and United States time zone references for information interchange. ANSI X3.51-1975; American National Standards Institute: New York, 1975.

Bhushan, A.K. The File Transfer Protocol. ARPANET Request for Comments, No. 354, Network Information Center No. 10596; Augmentation Research Center, Stanford Research Institute: Menlo Park, July 1972.

Bhushan, A.K. Comments on the File Transfer Protocol. ARPANET Request for Comments, No. 385, Network Information Center No. 11357; Augmentation Research Center, Stanford Research Institute: Menlo Park, August 1972.

Bhushan, A.K., Pogran, K.T., Tomlinson, R.S., and White, J.E. Standardizing Network Mail Headers. ARPANET Request for Comments, No. 561, Network Information Center No. 18516; Augmentation Research Center, Stanford Research Institute: Menlo Park, September 1973.

Feinler, E.J. and Postel, J.B. ARPANET Protocol Handbook. Network Information Center No. 7104; Augmentation Research Center, Stanford Research Institute: Menlo Park, April 1976. (NTIS AD A003890).

McKenzie, A. File Transfer Protocol. ARPANET Request for Comments, No. 454, Network Information Center No. 14333; Augmentation Research Center, Stanford Research Institute: Menlo Park, February 1973.

McKenzie, A. TELNET Protocol Specification. Network Information Center No. 18639; Augmentation Research Center, Stanford Research Institute: Menlo Park, August 1973.

Myer, T.H. and Henderson, D.A. Message Transmission Protocol. ARPANET Request for Comments, No. 680, Network Information Center No. 32116; Augmentation Research Center, Stanford Research Institute: Menlo Park, 1975.

Neigus, N. File Transfer Protocol. ARPANET Request for Comments, No. 542, Network Information Center No. 17759; Augmentation Research Center, Stanford Research Institute: Menlo Park, July 1973.

Pogran, K., Vittal, J., Crocker, D. and Henderson, A. Proposed official standard for the format of ARPA network messages.

ARPANET Request for Comments, No. 724, Network Information Center No. 37435; Augmentation Research Center, Stanford Research Institute: Menlo Park, May 1977.

Postel, J.B. Revised FTP Reply Codes. ARPANET Request for Comments, No. 640, Network Information Center No. 30843; Augmentation Research Center, Stanford Research Institute: Menlo Park, June 1974.